**PURBANCHAL UNIVERSITY**

****

**KHWOPA ENGINEERING COLLEGE**

**LIBALI-08, BHAKTAPUR**

LAB REPORT ON .NET

LAB NO. 01

**SUBMITTED BY: SUBMITTED TO:**

Name: Nischal Baidar Department of Computer Engineering

Roll No. : 770323

Group: A

Submission: 2081/12/08

**Theory:**

**Git** is a distributed version control system (DVCS) that allows multiple developers to work on a project simultaneously. It helps track changes, revert to previous versions, and manage collaboration efficiently.

**Features of Git:**

1. **Branching and Merging** – Work on different features without affecting the main codebase.
2. **History Tracking** – Maintain a log of all changes with commit messages.
3. **Collaboration** – Enables multiple developers to work on the same project without conflicts.

**GitHub**, on the other hand, is a cloud-based platform that hosts Git repositories, enabling collaboration. The workflow illustrated in the image represents how changes move through different stages in a Git project**.**

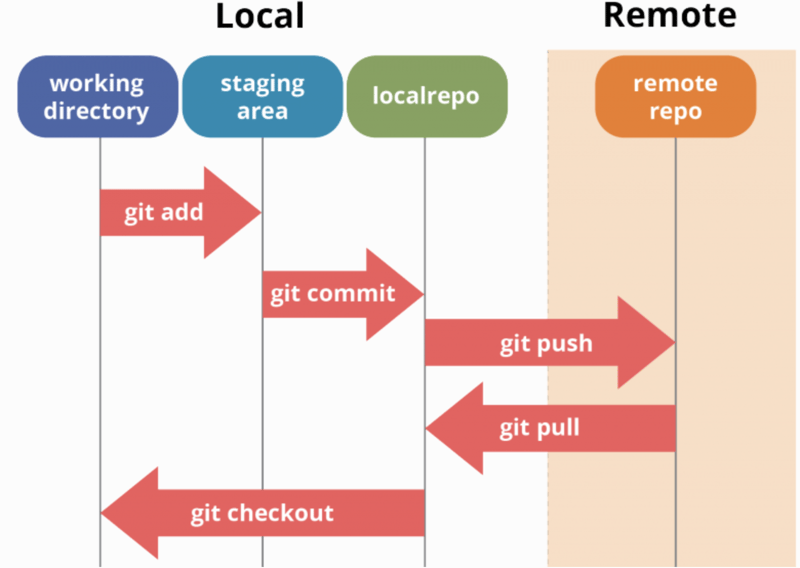
****

Fig: Git Workflow

**Local Workflow**

A Git project consists of three main areas:

1. Working Directory – This is where you modify files.
2. Staging Area – This holds changes that are marked for the next commit.
3. Local Repository – This store committed changes.

Key Commands:

* git add → Moves changes from the working directory to the staging area.
* git commit → Saves staged changes in the local repository.
* git checkout → Switches branches or restores files to a previous state.

**Syncing with Remote Repository**

The remote repository (hosted on GitHub) serves as a central location to share and collaborate on code.

Key Commands:

* git push → Uploads committed changes from the local repository to the remote repository.
* git pull → Fetches and integrates changes from the remote repository into the local repository.

By following this structured workflow, developers can efficiently manage code versions, collaborate with teams, and maintain a clean project history.

**Git and GitHub Commands Discussed during Lab Works:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Category** | | |  | | --- | | **Command** | | |  | | --- | | **Description** | |
| **Git Configuration** | git config --global user.name “nischal” | Sets your name for commits. |
|  | git config --global user.email “nischalbaidar@example.com” | Assigns an email to commits. |
| **Initializing a Repository** | git init | |  | | --- | |  |  |  | | --- | | Creates a new Git repository in the current directory. | |
| **Staging and committing** | git add . | |  | | --- | |  |  |  | | --- | | Stages all changes for commit. | |
|  | git commit -m “Commit message” | Saves staged changes with a message. |
| **Branching and Merging** | git branch | Lists all branches in the repository. |
|  | |  | | --- | |  |  |  | | --- | | git branch <branch\_name> | | |  | | --- | |  |  |  | | --- | | Creates a new branch. | |
|  | git checkout <branch\_name> / git switch <branch\_name> | Switches to another branch. |
|  | |  | | --- | |  |  |  | | --- | | git merge <branch\_name> | | Merges changes from one branch to another. |
| **Syncing with GitHub** | git push -u origin <branch\_name> | Uploads local commits to GitHub. |
|  | git pull origin <branch\_name> | Fetches and integrates changes from GitHub. |
| **Checking Status & History** | git status | Shows the current state of the working directory. |
|  | git log | Displays commit history |
| **Connecting to GitHub** | git remote add origin <repo\_url> | Links the local repository to GitHub. |

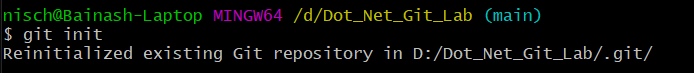
**Lab Works**

Configure the users username and email for commits.

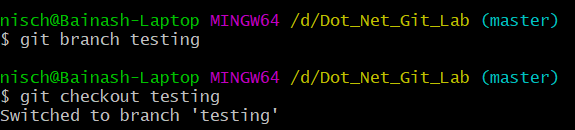




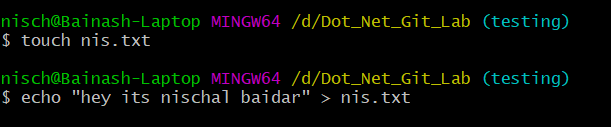
Initializing the git repository by creating a new folder.



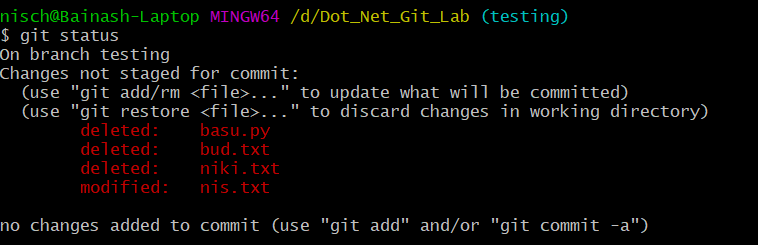
Create a new branch to work on and switch to it.



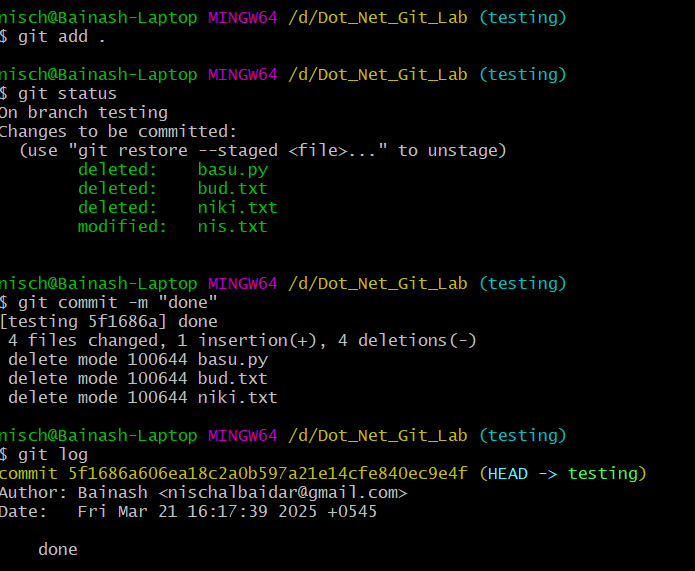
Create a new file and write content in it.

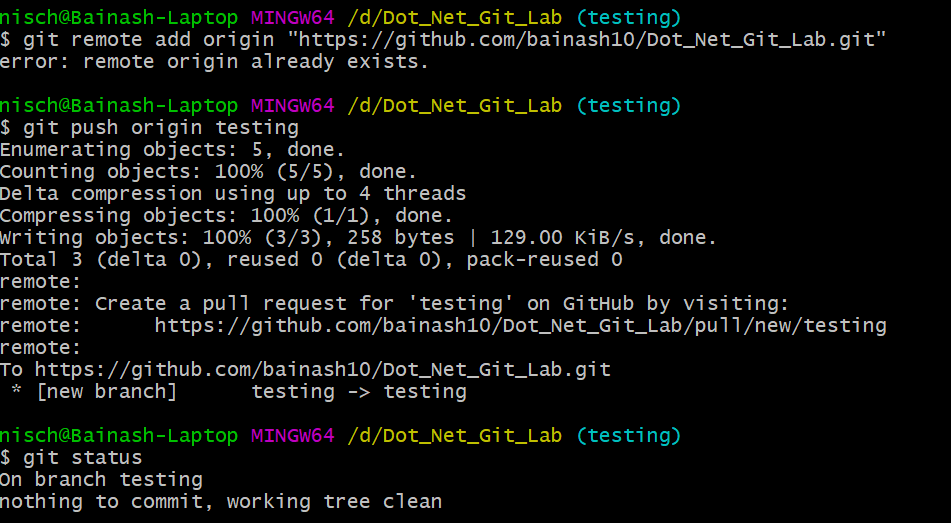


Now checking the status of testing branch. As testing branch is created from master branch so initially it had other files that was created during lab hours so I deleted it and added new text file nis.txt

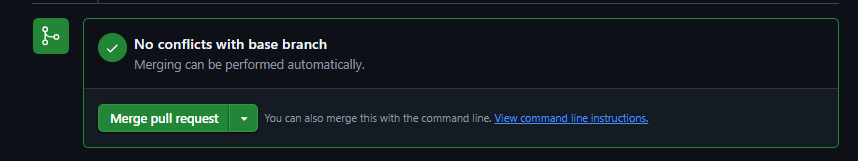


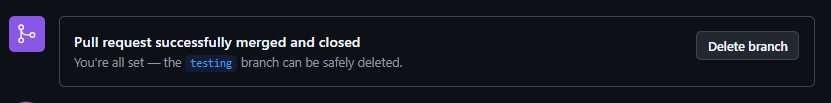
Now adding committing and syncing with GitHub:





Now merge the pull request by comparing master and testing branch





**Conclusion:**

Here we learned about how git works, why git is important and what is git.